Hopsworks

●●●

Dr. Jim Dowling
Sina Sheikholeslami

Nov 2019

Hopsworks Technical Milestones

World's fastest HDFS Published at USENIX FAST with Oracle and Spotify

World's first Hadoop platform to support GPUs-as-a-Resource

World's First Open Source Feature Store for Machine Learning

First non-Google ML Platform with TensorFlow Extended (TFX) support through Beam/Flink

2017

2018

2019

Winner of IEEE Scale Challenge 2017 with HopsFS - 1.2m ops/sec

World's First Distributed Filesystem to store small files in metadata on NVMe disks

World's most scalable POSIX-like Hierarchical Filesystem with Multi Data Center Availability with 1.6m ops/sec on GCP

World's First Unified Hyperparam and Ablation Study Parallel Prog.` Framework

"If you're working with big data and Hadoop, **this one paper could repay your investment** in the Morning Paper many times over.... **HopsFS is a huge win.**"

- Adrian Colyer, The Morning Paper

0. Slides:
http://hops.io/id2223.pdf

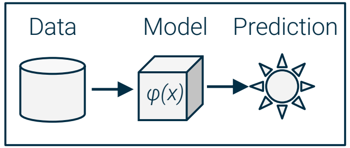
1. Register for an account at:
www.hops.site

2. Follow the Instructions here:
https://bit.ly/2UEixTr

3. Getting started Videos
https://bit.ly/2NnbKgu

Hopsworks hides the Complexity of Deep Learning

Hopsworks
Feature Store



Hopsworks
REST API

[Adapted from Schulley et Al "Technical Debt of ML"]

Datasources

Hopsworks

The Platform for Data Intensive AI

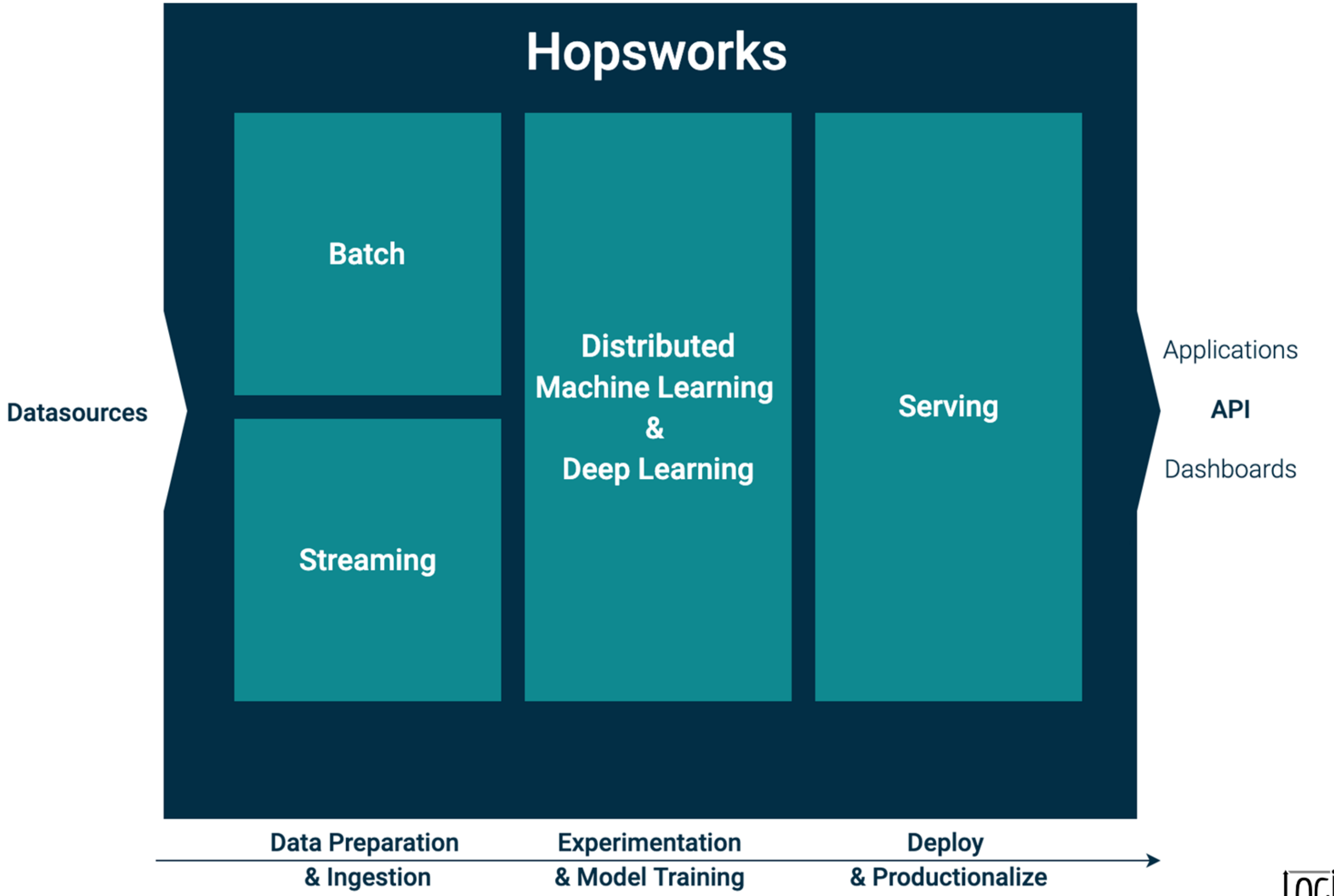
-

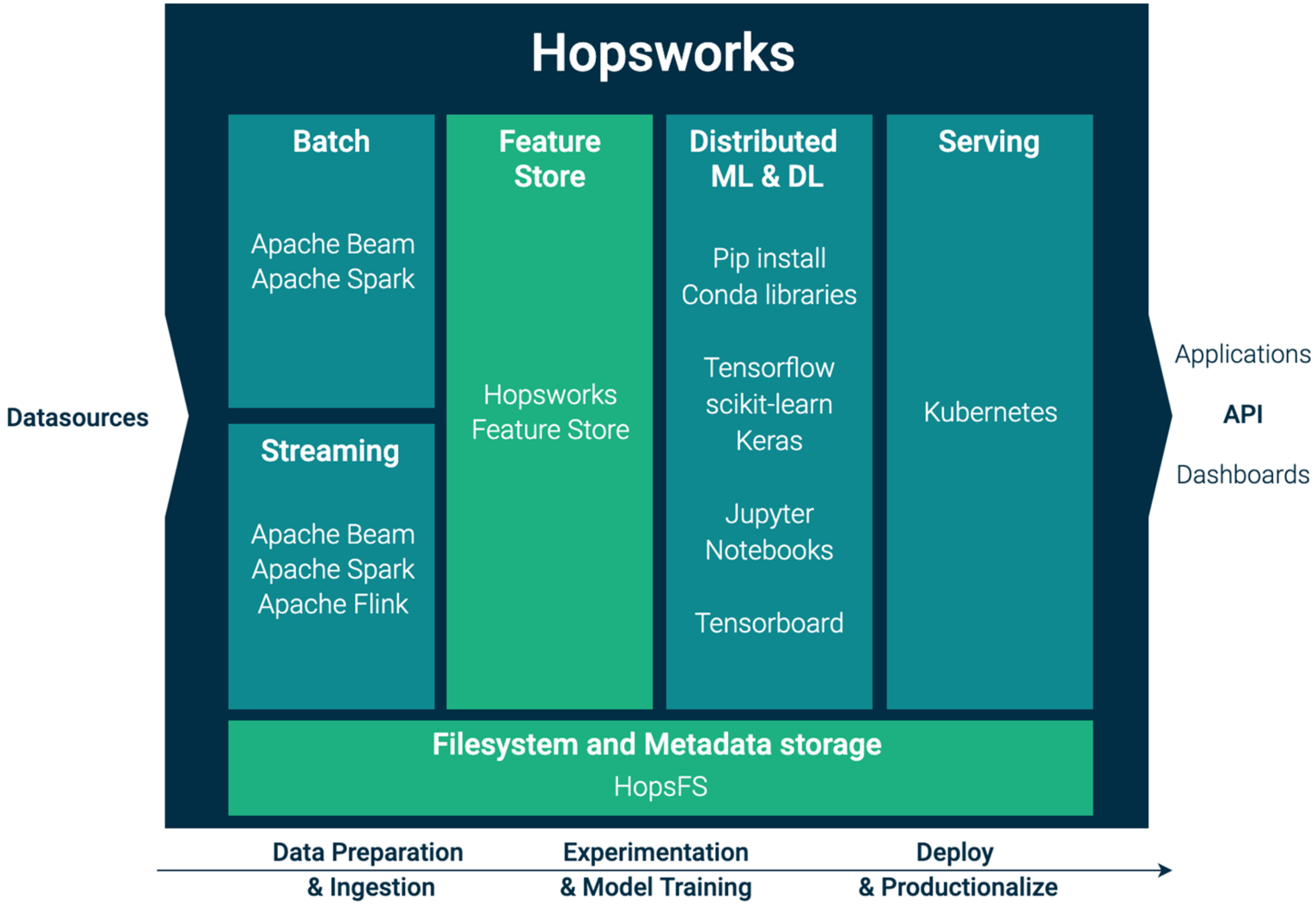
Machine Learning, Deep Learning & Model serving

Applications

API

Dashboards

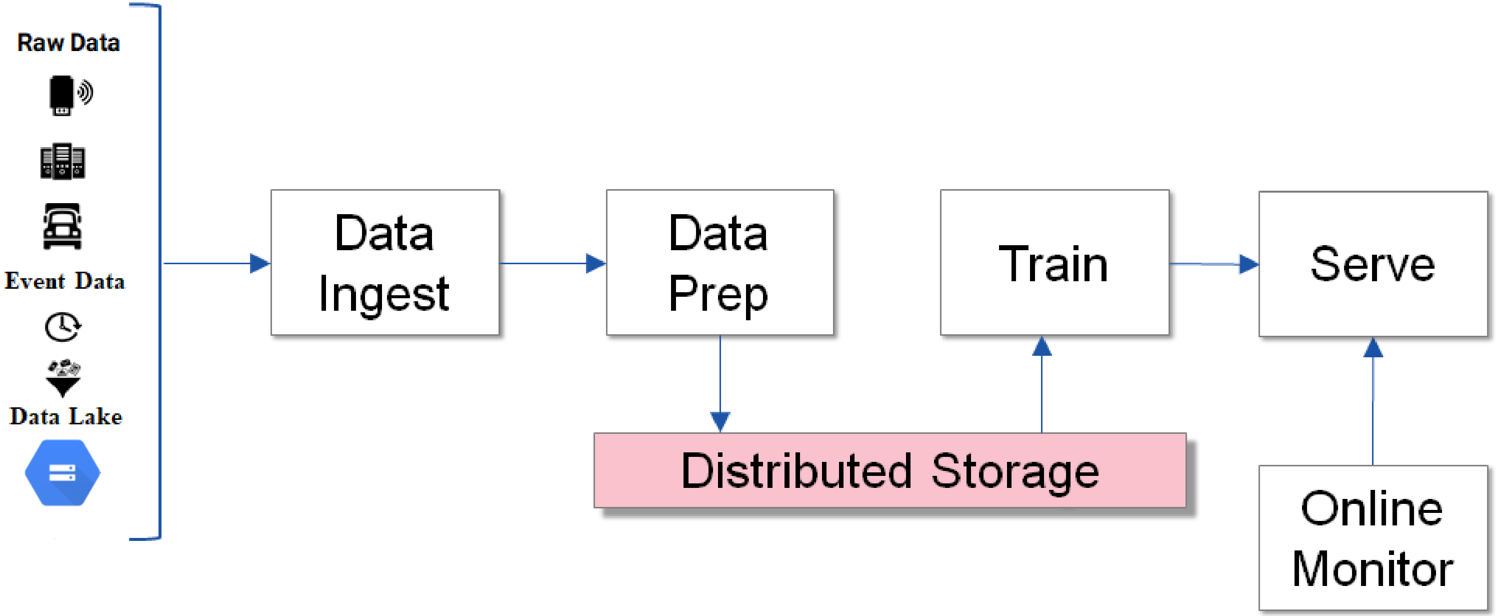

Hopsworks
Batch
Streaming
Distributed
Machine Learning
&
Deep Learning
Serving
Datasources
Applications
API
Dashboards
Data Preparation
& Ingestion
Experimentation
& Model Training
Deploy
& Productionalize
LOGICAL CLOCKS



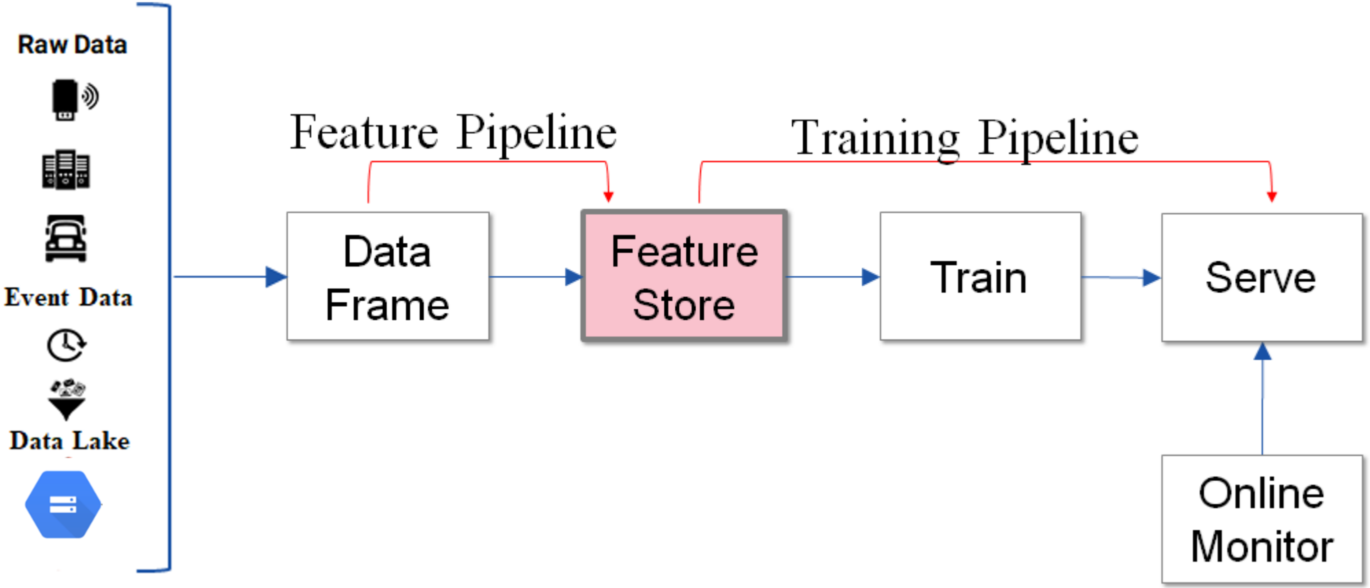
Hopsworks
Batch
Apache Beam
Apache Spark
Streaming
Apache Beam
Apache Spark
Apache Flink
Feature Store
Hopsworks Feature Store
Distributed ML & DL
Pip install
Conda libraries
Tensorflow
scikit-learn
Keras
Jupyter Notebooks
Tensorboard
Serving
Kubernetes
Filesystem and Metadata storage
HopsFS
Datasources
Applications
API
Dashboards
Data Preparation & Ingestion
Experimentation & Model Training
Deploy & Productionalize

Machine Learning Pipelines

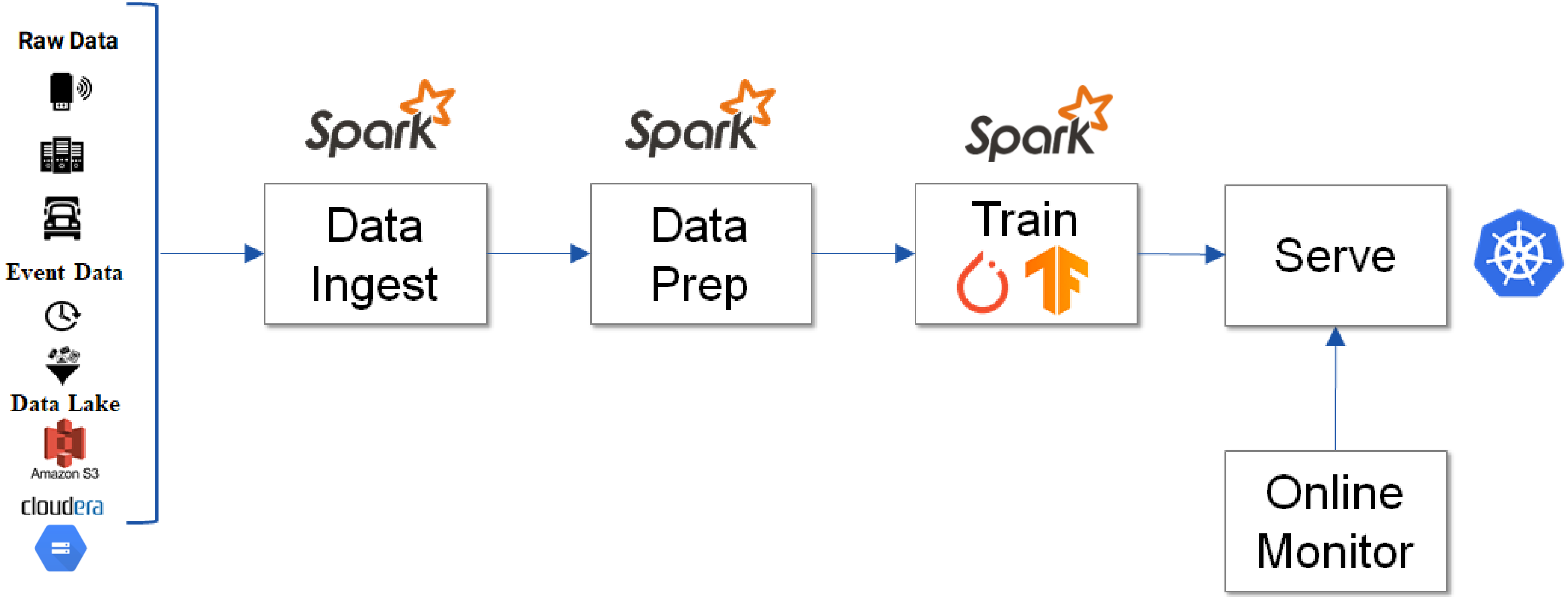
End-to-End ML Pipelines



ML Pipelines with a Feature Store



End-to-End ML Pipelines in Hopsworks



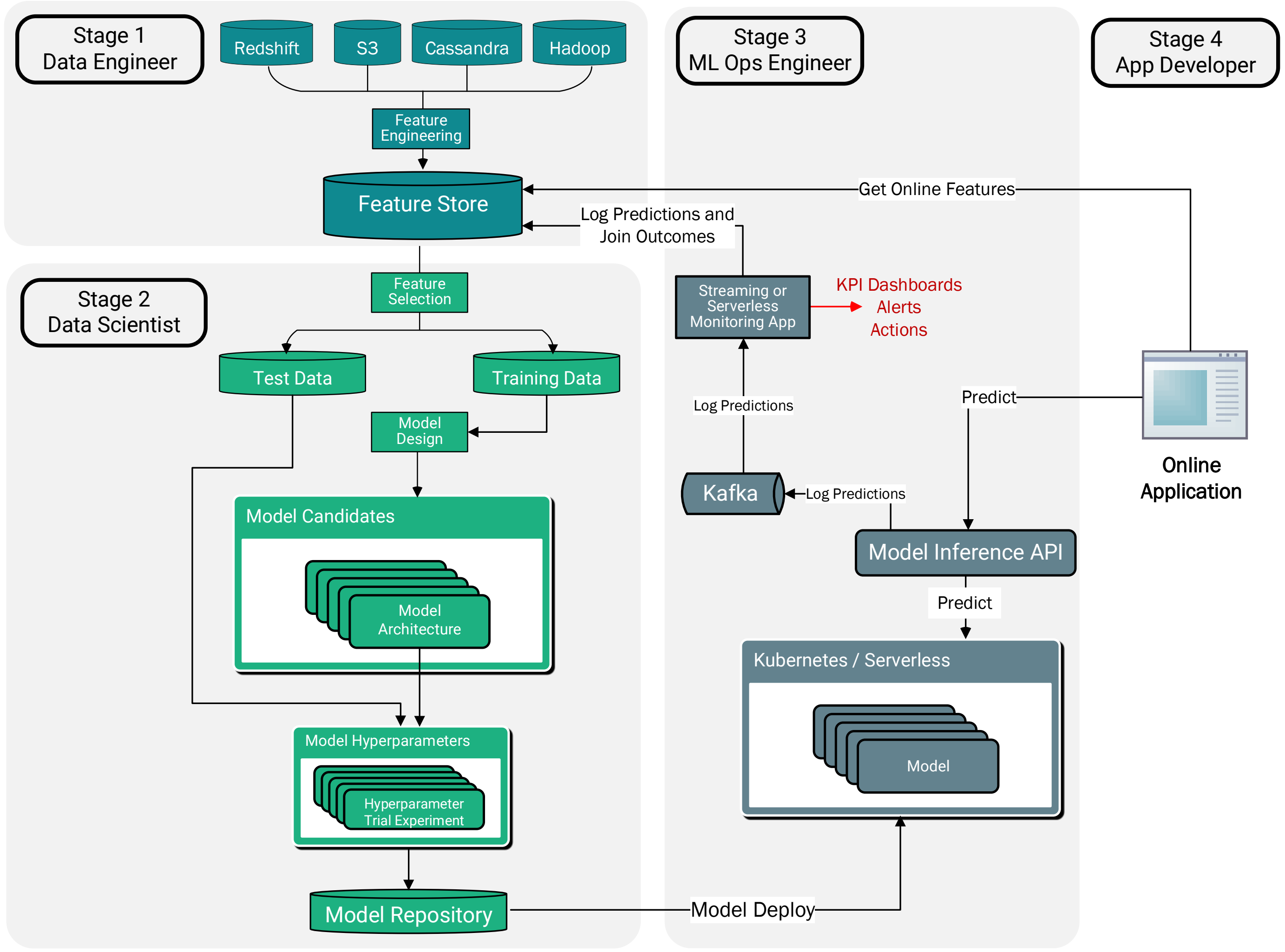
Roles in Machine Learning

Stage 1. Data Engineer

Stage 2. Data Scientist

Stage 3. ML Ops Engineer

Stage 4. App Developer

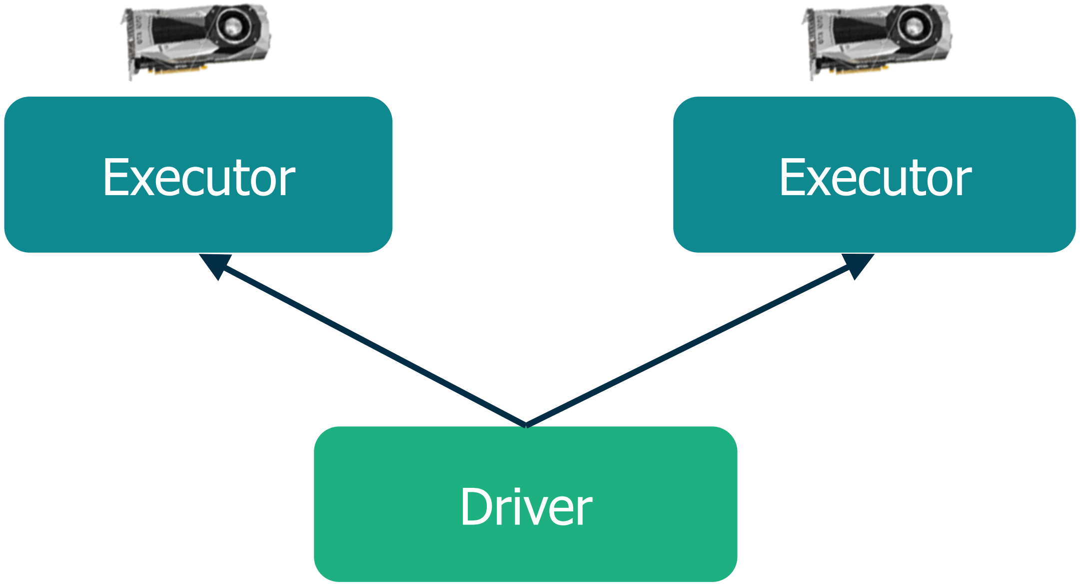


Running TensorFlow/Keras/PyTorch Apps in PySpark

Warning: micro-exposure to PySpark may cure you of distributed programming phobia

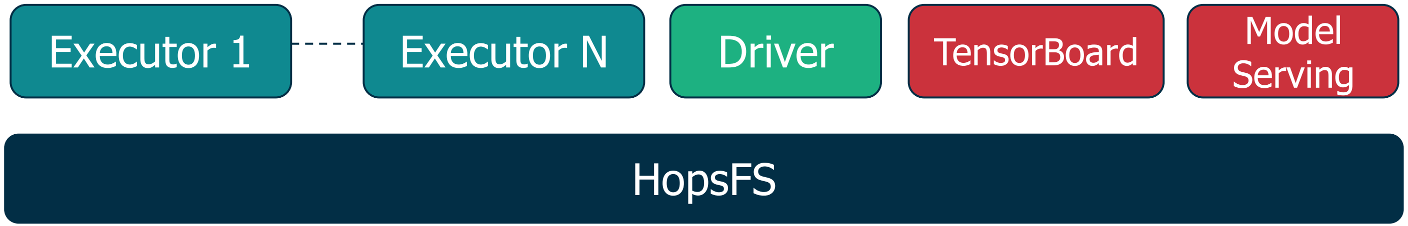
GPU(s) in PySpark Executor, Driver coordinates

PySpark makes it easier to write TensorFlow/Keras/ PyTorch code that can either be run on a single GPU or scale to run on lots of GPUS for Parallel Experiments or Distributed Training.



Need Distributed Filesystem for Coordination

- Training/Test Datasets
- Model checkpoints, Trained Models
- Experiment run data
- Provenance data
- Application logs



PySpark Hello World

```python
def executor():
  print(”Hello from GPU”)

from hops import experiment
experiment.launch(executor)
```

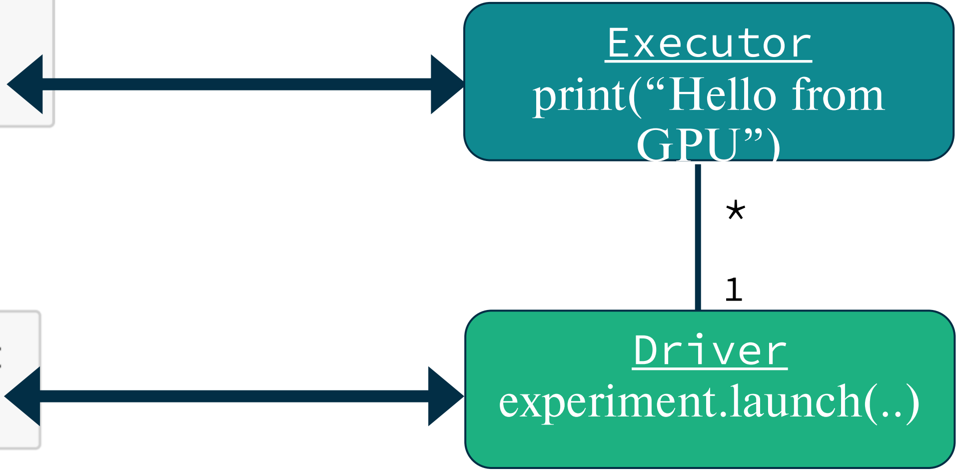
PySpark – Hello World

In [ ]:

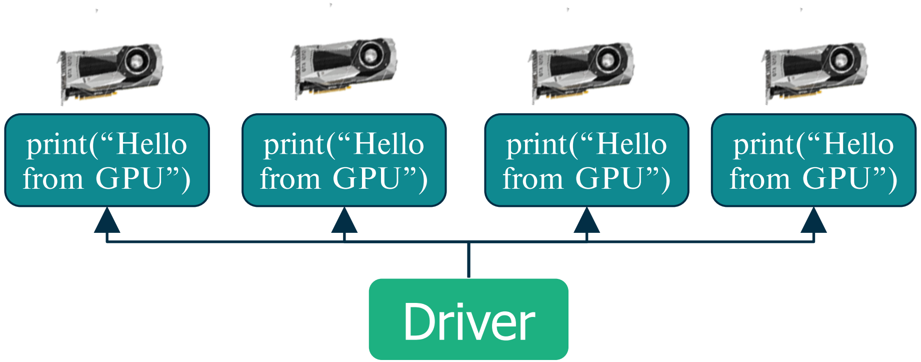
```
def executor():
  print(”Hello from GPU”)
```

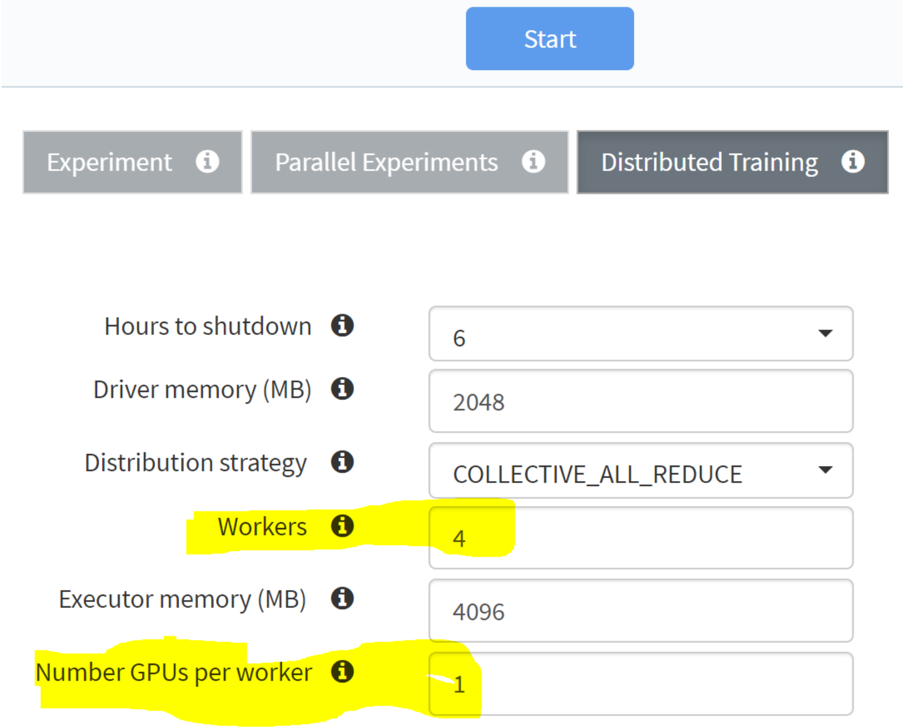
In [ ]:

```
from hops import experiment
experiment.launch(executor)
```



Leave code unchanged, but configure 4 Executors





Driver with 4 Executors

In [ ]:
```
def executor():
  print(”Hello from GPU”)
```

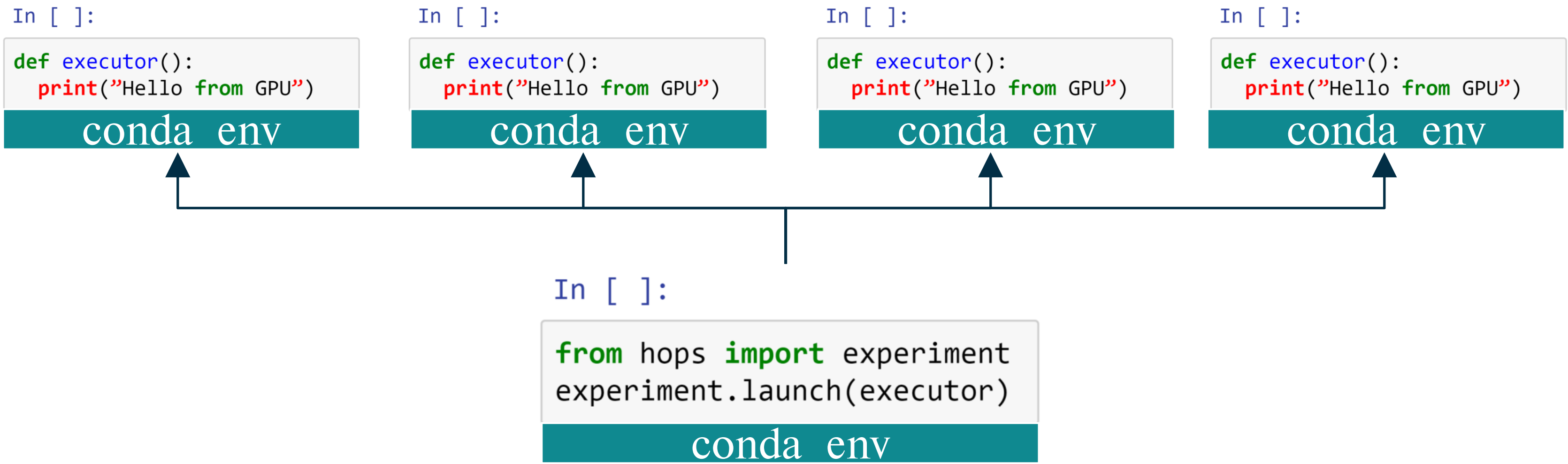
In [ ]:
```
def executor():
  print(”Hello from GPU”)
```

In [ ]:
```
def executor():
  print(”Hello from GPU”)
```

In [ ]:
```
def executor():
  print(”Hello from GPU”)
```

In [ ]:
```
from hops import experiment
experiment.launch(executor)
```

Same/Replica Conda Environment on all Executors



A Conda Environment Per Project in Hopsworks

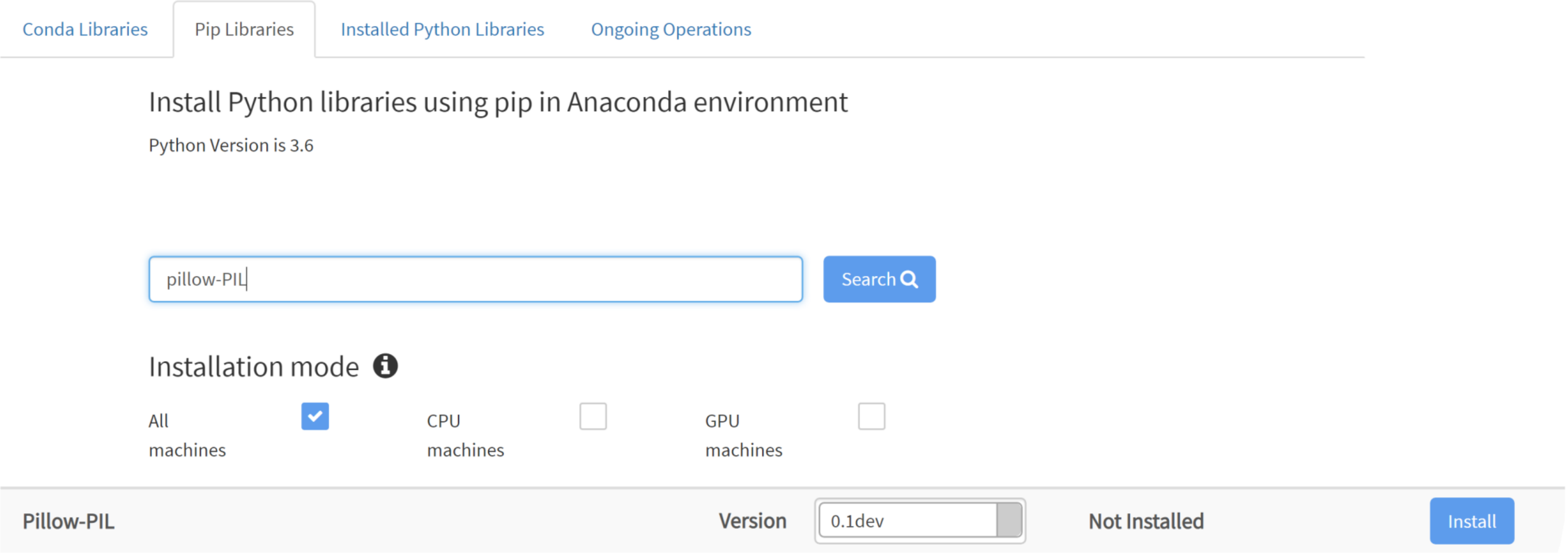
Conda Libraries | Pip Libraries | Installed Python Libraries | Ongoing Operations | See hops python docs

Uninstall/Upgrade Python Libraries

Export Environment

Url	Library	Version	Package Manager	MachineType	Status	User-Installed
PyPi	hopsfacets	0.0.3	PIP	ALL	SUCCESS	Uninstall
PyPi	pandas	0.23.1	PIP	ALL	SUCCESS	Uninstall
PyPi	mmlspark	0.13	PIP	ALL	SUCCESS	Uninstall
PyPi	numpy	1.15.3	PIP	ALL	SUCCESS	Uninstall
PyPi	hops	0.6.0.1	PIP	ALL	SUCCESS	Uninstall
PyPi	pydoop	2.0a3	PIP	ALL	SUCCESS	Pre-installed
PyPi	tensorboard	1.11.0	PIP	ALL	SUCCESS	Pre-installed
PyPi	tensorflow	1.11.0	PIP	CPU	SUCCESS	Pre-installed
PyPi	tensorflow-gpu	1.11.0	PIP	GPU	SUCCESS	Pre-installed

Use Pip or Conda to install Python libraries



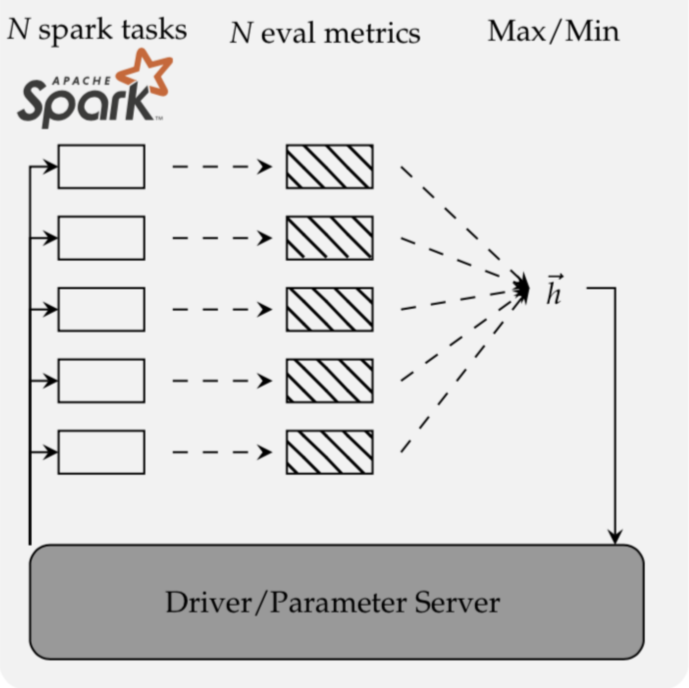
LOGICAL CLOCKS

TensorFlow Distributed Training with PySpark

```
def train():
  # Separate shard of dataset per worker
  # create Estimator w/ DistribStrategy
  # as CollectiveAllReduce
  # train model, evaluate
  return loss

# Driver code below here
# builds TF_CONFIG and shares to workers
from hops import experiment
experiment.collective_allreduce(train)
```

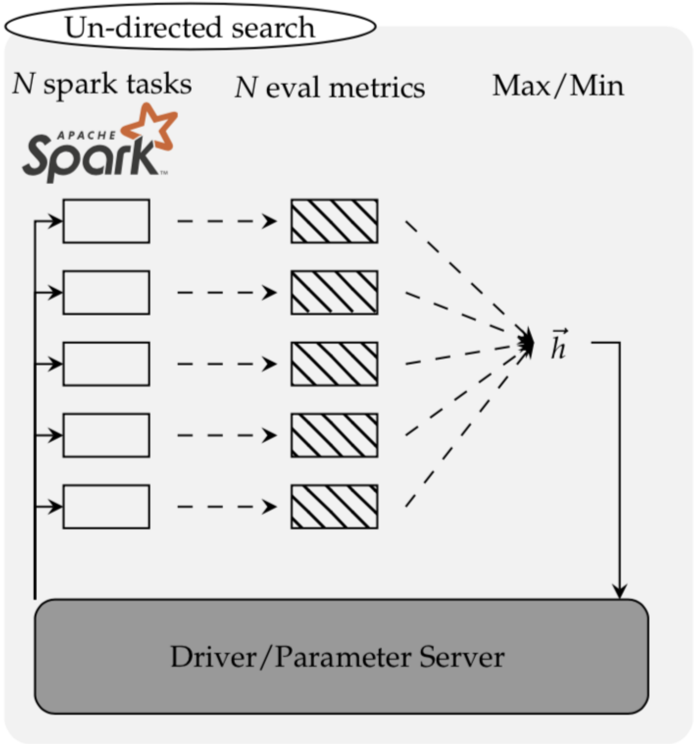
More details: http//github.com/logicalclocks/hops-examples



Undirected Hyperparam Search with PySpark

```
def train(dropout):
  # Same dataset for all workers
  # create model and optimizer
  # add this worker's value of dropout
  # train model and evaluate
  return loss

# Driver code below here
from hops import experiment
args={"dropout":[0.1, 0.4, 0.8]}
experiment.grid_search(train,args)
```

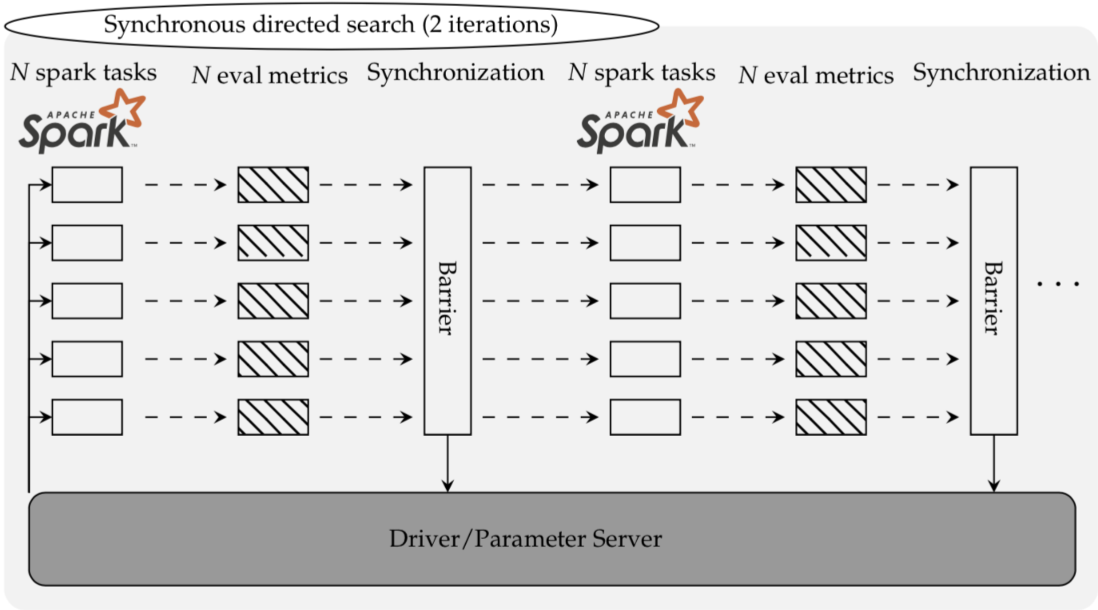


More details: http//github.com/logicalclocks/hops-examples

Directed Hyperparameter Search with PySpark

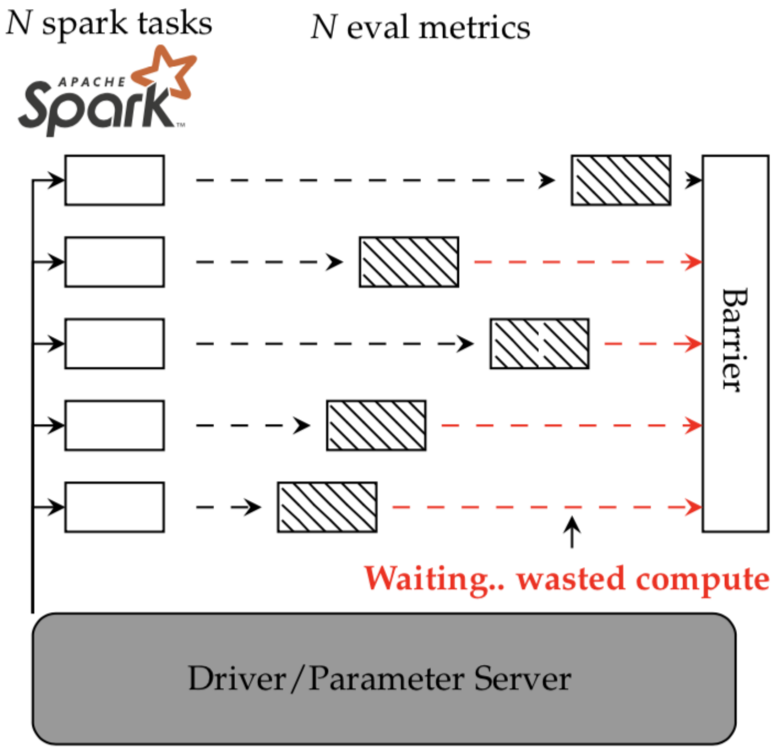
```
def train(dropout):
  # Same dataset for all workers
  # create model and optimizer
  optimizer.apply(dropout)
  # train model and evaluate
  return loss

from hops import experiment
args={“dropout”: “0.1-0.8”}
experiment.diff_ev(train,args)
```



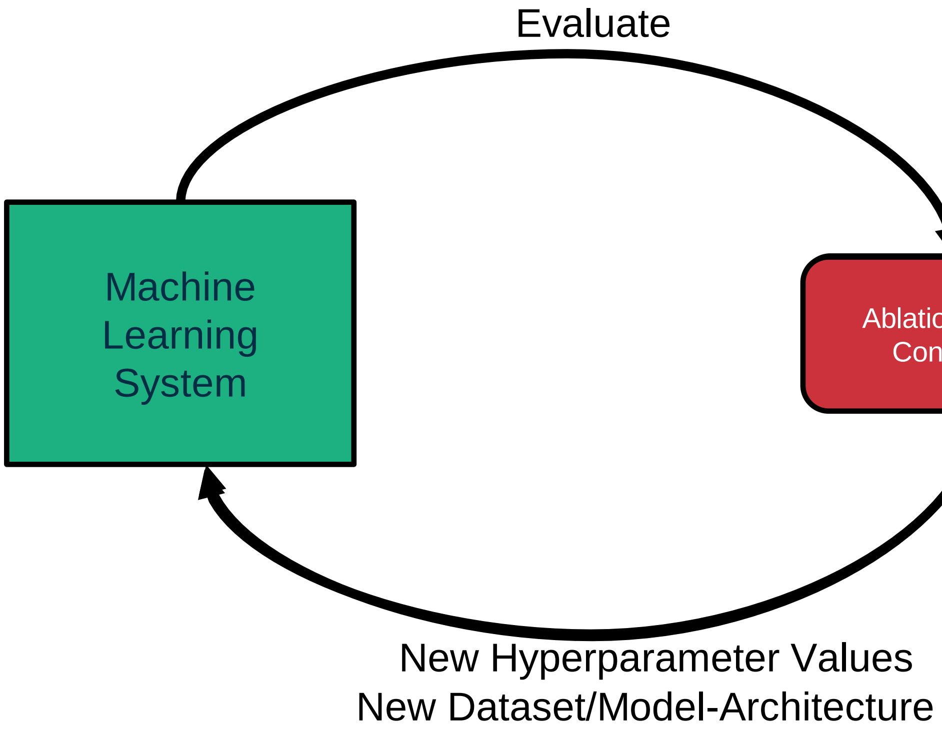
More details: http//github.com/logicalclocks/hops-examples

Wasted Compute!



Parallel ML Trials with Maggy

Maggy: Unified Hparam Opt & Ablation Programming



Synchronous or
Asynchronous Trials

Directed or
Undirected Search

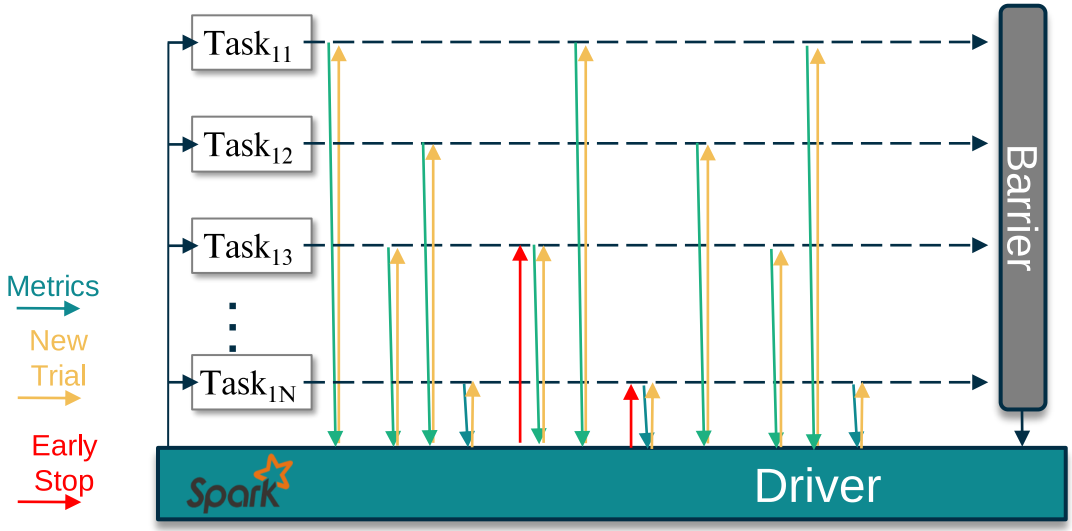
User-Defined
Search/Optimizers

Directed Hyperparameter Search with Maggy

```
def train(dropout, reporter):
 …..

from maggy import experiment,
Searchspace
sp =
SearchSpace(dropout=('INTEGER',
[2, 8]))

experiment.lagom(train, sp)
```

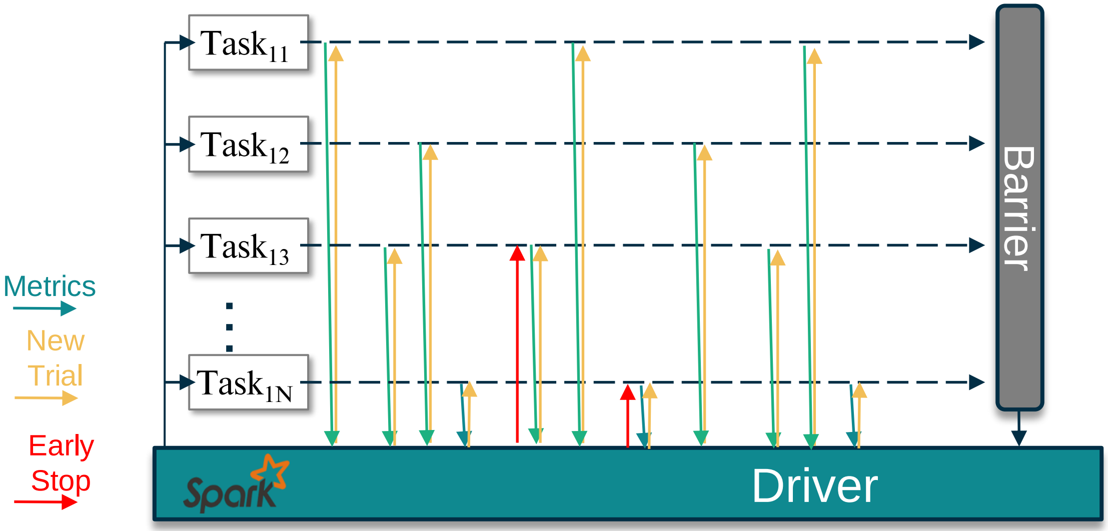


More details: http//github.com/logicalclocks/hops-examples

Parallel Ablation Studies with Maggy

```
def train(dataset_function,
model_function):
 …..


from maggy import experiment
ablation_study=…
experiment.lagom(train,
experiment_type='ablation',
ablation_study=ablation_study,
ablator='loco')
```



More details: http//github.com/logicalclocks/hops-examples

/Experiments

- Executions add entries in /Experiments:

```
experiment.launch(…)
experiment.grid_search(…)
experiment.collective_allreduce(…)
experiment.lagom(…)
```

- /Experiments contains:
  - logs (application, tensorboard)
  - executed notebook file
  - conda environment used
  - checkpoints

```
/Projects/MyProj
      └ Experiments
         └ <app_id>
            └ <type>
               ├─ checkpoints
               ├─ tensorboard_logs
               ├─ logfile
               └─ versioned_resources
                  ├─ notebook.ipynb
                  └─ conda_env.yml
```

/Models

- Named/versioned model management for:
  TensorFlow/Keras
  Scikit Learn
- A `Models` dataset can be securely shared with other projects or the whole cluster
- The provenance API returns the `conda.yml` and `execution` used to train a given model

```
/Projects/MyProj
         └ Models
             └ <name>
                 └ <version>
                     ├─ saved_model.pb
                     └─ variables/
                      ...
```

That was Hopsworks

Efficiency & Performance

Feature Store
Data warehouse for ML

Distributed Deep Learning
Faster with more GPUs

HopsFS
NVMe speed with Big Data

Horizontally Scalable
Ingestion, DataPrep,
Training, Serving

Development & Operations

Development Environment
First-class Python Support

Version Everything
Code, Infrastructure, Data

Model Serving on Kubernetes
TF Serving, SkLearn

End-to-End ML Pipelines
Orchestrated by Airflow

Security & Governance

Secure Multi-Tenancy
Project-based restricted access

Encryption At-Rest, In-Motion
TLS/SSL everywhere

AI-Asset Governance
Models, experiments, data, GPUs

Data/Model/Feature Lineage
Discover/track dependencies

Acknowledgements and References

Slides and Diagrams from colleagues:

- Maggy: Moritz Meister and Sina Sheikholeslami
- Feature Store: Kim Hammar
- Beam/Flink on Hopsworks: Theofilos Kakantousis

References

- HopsFS: Scaling hierarchical file system metadata …, USENIX FAST 2017.
- Size matters: Improving the performance of small files …, ACM Middleware 2018.
- ePipe: Near Real-Time Polyglot Persistence of HopsFS Metadata, CCGrid, 2019.
- Hopsworks Demo, SysML 2019.

Thank you!

470 Ramona St

Palo Alto

https://www.logicalclocks.com

Register for a free account at

www.hops.site

Twitter

@logicalclocks

@hopsworks

GitHub

https://github.com/logicalclocks/hopsworks

https://github.com/hopshadoop/hops